TRANSMITTAL OF APPEAL BRIEF (Large Entity)	Docket No. PM 2000.063

OIPE IAP22 NOV 02 2007 PATENT & TRADEMARK OFFICE

In Re Application Of: **Attila Banki et al.**

Application No.	Filing Date	Examiner	Customer No.	Group Art Unit	Confirmation No.
10/020,033	December 6, 2001	Jason Scott Proctor	3477	2123	8954

Invention: **COMPUTER SYSTEM AND METHOD HAVING A FACILITY MANAGEMENT LOGIC ARCHITECTURE**

COMMISSIONER FOR PATENTS:

Transmitted herewith is the Appeal Brief in this application, with respect to the Notice of Appeal filed on:

The fee for filing this Appeal Brief is: **$510.00**

☐ A check in the amount of the fee is enclosed.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 05-1328. I have enclosed a duplicate copy of this sheet.

☐ Payment by credit card. Form PTO-2038 is attached.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Gary D. Lawson

Signature

Dated: **30 October 2007**

Gary D. Lawson, Reg. No. 27,696
ExxonMobil Upstream Research Company
P. O. Box 2189
(CORP-URC-SW341)
Houston, Texas 77252-2189

713-431-4846 Phone
713-431-4664 Facsimile

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on

30 October 2007

(Date)

Monica Stansberry

Signature of Person Mailing Correspondence

Monica Stansberry

Typed or Printed Name of Person Mailing Correspondence

cc: **DLP, TEM**

Serial No. 10/020,033

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

IN RE APPLICATION OF ATTILA BANKI ET AL.

"COMPUTER SYSTEM AND METHOD HAVING A FACILITY MANAGEMENT LOGIC ARCHITECTURE"

On Appeal from the Office Action of the Examiner Mailed April 13, 2007, finally rejecting all pending claims

Examiner Jason Scott Proctor
Group Art Unit 2123

APPEAL BRIEF

Gary D. Lawson, Reg. No. 27,696
ExxonMobil Upstream Research Company
P. O. Box 2189
Houston, Texas 77252-2189
Telephone: (713) 431-4846
Facsimile: (713) 431-4664

ATTORNEY FOR APPELLANT

TABLE OF CONTENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of: Attila Banki et al.	§	Confirmation No.: 8954
Serial No. 10/020,033	§	Examiner: Jason Scott Proctor
Filed: December 6, 2001	§	Art Unit: 2123
Title: "COMPUTER SYSTEM AND METHOD HAVING A FACILITY MANAGEMENT LOGIC ARCHITECTURE"	§	

APPEAL BRIEF

MS: Appeal Briefs - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

This Appeal Brief is submitted in support of the Appeal in the above-identified application.

Certificate of Mailing Under 37 C.F.R. §1.8(a)

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Briefs - Patents, Commissioner for Patents, Alexandria, VA 22313-1450 on October 30, 2007.

Monica Stansberry
Monica Stansberry

REAL PARTY IN INTEREST

The real party in interest in the present application is the Assignee, ExxonMobil Upstream Research Company, as evidenced by the Assignment recorded on December 6, 2001 at Reel 012388 / Frame 0915.

RELATED APPEAL AND INTERFERENCES

There are no Appeals or Interferences known to Appellant, the Appellant's legal representative, or assignee, which directly affect or would be directly affected by or have a bearing on the Board's decision in the pending appeal.

STATUS OF THE CLAIMS

Claims 1-13, 15-28, 30-31 and 43-46 are pending. All pending claims stand finally rejected as noted in the Examiner's Action mailed April 13, 2007. In the present paper, rejected Claims 1-13, 15-28, 30-31 and 43-46 are appealed. All pending claims are listed in the Claims Appendix.

STATUS OF AMENDMENTS

No amendment has been submitted subsequent to the final rejection in the Office Action mailed April 13, 2007.

SUMMARY OF CLAIMED SUBJECT MATTER

A computer system of some embodiments enables a simulator user to simulate transport phenomena (e.g., page 7, line 27 to page 8, line 11) in a physical system. The computer system comprises a processor, memory coupled to the processor and object-oriented software (e.g., page 9 beginning with line 21) in a main simulation system (e.g., page 23, line 1-7) stored in the memory (e.g., page 7, line 20). The object-oriented software is configured to provide a logic interface (e.g., page 17, lines 31-32) to dynamically construct logic (e.g., page 18, lines 4-6) to customize

simulation of transport phenomena through a model of the physical system (e.g., page 18, lines 4-6). The object-oriented software is further configured to convert the dynamically construct logic into corresponding object-oriented code during a simulation without intervention of the simulator user (e.g., page 21, lines 6-7). The object-oriented software is capable of being integrated, without intervention of the simulator user (e.g., page 25, lines 19-21), with the main simulation system which comprises a simulation data model and simulation algorithms (e.g., page 10, lines 29-31), resulting in an integrated simulation system (e.g., page 25, lines 23-24). The object-oriented code thereby extends (e.g., page 6, lines 20-22 and page 25, lines 15-18) the simulation data model by creating new classes (e.g., page 25, lines 18-19) that inherit from the simulation data model. The object-oriented code is also configured to call functions (e.g., page 10, lines 5-6 and page 26, lines 12-22) of the integrated simulation system and use member data (e.g., page 10 lines 5-6) of the integrated simulation system and to execute (e.g., page 18, lines 26-28) the integrated simulation system.

The method is particularly useful in simulating a reservoir system. In a reservoir system model, a facility network model ("FNM") is preferably developed that extends the discretized reservoir simulation model (consisting of nodes and flow connections) beyond the reservoir to include nodes and connections for modeling fluid flow in the well tubulars and surface production and injection facilities (such as manifolds, pumps, compressors, gathering lines, separators and pipelines). Each distinct facility type is modeled as a specialized type of node, connection, or a composite of several nodes and connections. In its most basic form, the entire simulation model can encompass every component of the hydrocarbon field from the subsurface reservoir, all wells and well hardware, and surface facilities up to and including the product delivery outlet(s) (e.g., page 6, lines 20-30).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-13, 15-18, 20, 23-28, 30, 43, and 46 are rejected under 35 U.S.C. §103(a) as allegedly unpatentable over "Real-Time Workshop" (referred to by the herein as "*Real-Time Workshop*") in view of "The C++ Programming Language" by Bjarne Stroustrup (referred to herein as "*Stroustrup*").

Claims 21-22 and 44-45 are rejected under 35 U.S.C. §103(a) as purportedly unpatentable over *Real-Time Workshop* in view of *Stroustrup* and further in view of U.S. patent 6,052,520 ("*Watts*").

Claims 19 and 31 are rejected under 35 U.S.C. §103(a) as purportedly unpatentable over *Real-Time Workshop* in view of *Stroustrup* and further in view of Official Notice.

ARGUMENT

Claim 1 presently stands rejected in the final Office Action under 35 U.S.C. §103(a) as allegedly unpatentable over "Real-Time Workshop" (referred to herein as "*Real-Time Workshop*") in view of "The C++ Programming Language" by Bjarne Stroustrup (referred to herein as "*Stroustrup*").

As a preliminary matter, in the final Office Action, the Examiner referred to *Real-Time Workshop* as "Simulink", apparently believing that *Real-Time Workshop* is made by Simulink. However, *Real-Time Workshop* is owned by The Math Works Inc. and is used with Simulink, another product by The Math Works. Because Simulink and *Real-Time Workshop* are separate products, to avoid confusion, the document entitled *Real-Time Workshop* is referred to herein by that name. Page xvi of *Real-Time Workshop* shows a schematic of the relationship between Simulink and *Real-Time Workshop*. Applicants' comments herein regarding the functionality of Simulink are limited to the disclosure in *Real-Time Workshop*. The relationship between Simulink and *Real-Time Workshop* is discussed in more detail below.

As a second preliminary matter, in the Advisory Action, the Examiner made specific assertions that the Applicants believe should be clarified. To begin, the Examiner acknowledged in both the final Office Action and the Advisory Action that

the cited references do not recite the claim language verbatim. In the Advisory Action, the Examiner states that "a person of ordinary skill in the art of modeling and simulation, in light of that person's knowledge of computer based simulation and modeling software as evidenced by the references of record in this application and especially in light of specific references applied under 35 USC 103 and for the explicit rationale set forth in the Office Action would have found the claimed invention obvious." That is, the Examiner appears to assert that each of the claim elements does not have to be shown in the prior art references.

The Examiner asserts that a person skilled in the art would have been motivated to combine the object-oriented principles in *Stroustrup* with the non-objected-oriented principles in *Real-Time Workshop* because the "traditional design methods (non-object-oriented principles are 'less resilient to change, less amendable to tools, less suited for parallel development, and less suited for concurrent execution'" (final Office Action page 6).

Applicants respectfully traverse the Examiner's allegations.

"To establish *prima facie* obviousness of a claimed invention, **all the claim limitations** must be taught or suggested by the prior art." M.P.E.P. § 2143.03 (emphasis added); see also *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974). "**All words** in a claim must be considered in judging the patentability of that claim against the prior art." M.P.E.P. 2143.03 (emphasis added); see also *In re Wilson*, 424 F.2d 1382, 1385, 165 U.S.P.Q. 494, 496 (C.C.P.A. 1970). Applicants respectfully submit that each and every element of pending claims 1-13, 15-28, 30-31 and 43-46 is not found within the references cited by the Examiner.

To reiterate certain aspects of the Applicants' invention prior to addressing the references cited in the final Office Action, claim 1 is discussed below as an exemplary claim for discussion. Claim 1 presently reads as follows:

1. A computer system for simulating a physical system comprising:

   a processor;

   memory coupled to the processor; and

object-oriented software in a main simulation system stored in the memory, the object-oriented software configured to:

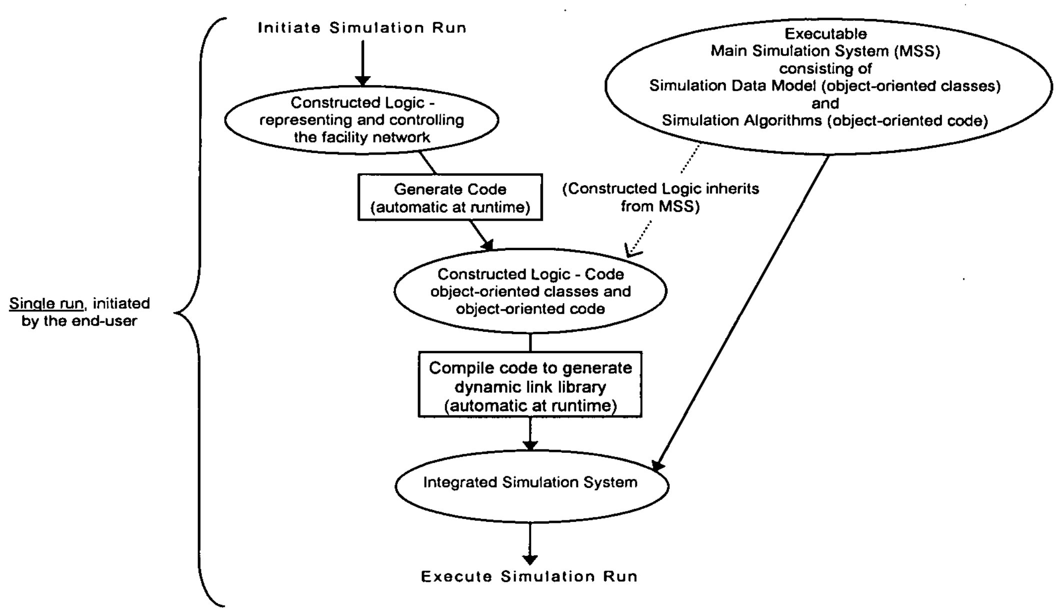
a) provide a logic interface to dynamically construct logic to customize simulation of transport phenomena through a model of the physical system;

b) convert the constructed logic into corresponding object-oriented code during a simulation without intervention of the simulator user;

c) integrate, without intervention of the simulator user, the object-oriented code with the main simulation system which comprises a simulation data model and simulation algorithms, resulting in an integrated simulation system, wherein the object-oriented code extends the simulation data model by creating new classes that inherit from the simulation data model, and the object-oriented code is configured to call functions of the integrated simulation system and use member data of the integrated simulation system; and

d) execute the integrated simulation system. (emphasis added)

The following schematic generally shows the relationship of constructed logic, the simulation model, and integrated simulation system of Applicants' claimed computer system.

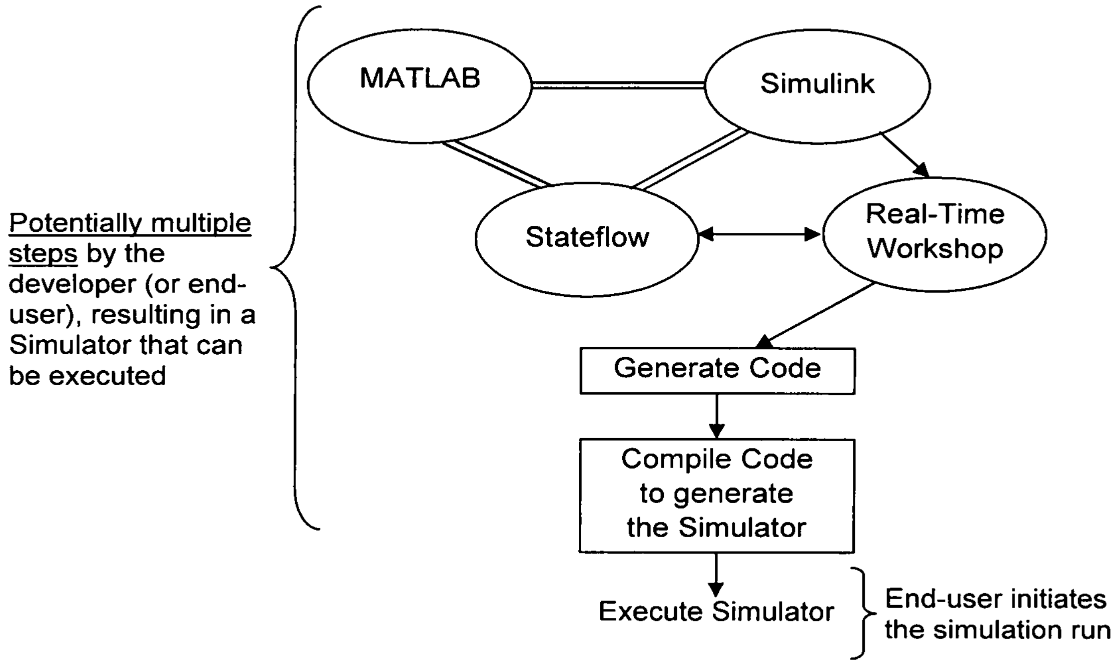


As shown in the schematic, the Applicants' claimed system performs simulation steps all during one single run. The Constructed Logic, which represents and controls a model of the facility network, is at runtime converted into sophisticated C++ object-oriented code (using principles of inheritance, encapsulation, polymorphism, and multiple inheritance). This code is automatically constructed in a manner such that it seamlessly <u>extends</u> the Simulation Data Model that exists in the executable Main Simulation System, such that the Constructed Logic is able to use the data and methods of the Main Simulation System, in addition to extending the data model and customizing the methods. The Constructed Logic code is then, again during runtime, compiled. The resulting dynamic link library embodies the new data model and algorithms that complement those in the Main Simulation System to form the user-customized "Integrated Simulation System".

Applicants' computer system is particularly advantageous in enabling a user to simulate a hydrocarbon-bearing reservoir associated with production facilities. Applicants' computer system provides an intuitive tool to capture a simulator user's custom facility management logic (the physical system), turn that into executable code that is integrated with the reservoir simulator's code (the main simulation system), and then execute the resulting system to predict the overall behavior of the reservoir during its lifetime (see page 5, lines 9-12 of the application).

Real-Time Workshop discloses a software package for modeling, simulating and analyzing dynamic systems utilizing a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. While *Real-Time Workshop* discloses a graphical user interface (GUI) for building models that "can run in a variety of environments, including real-time and stand-alone simulations," there is nothing in *Real-Time Workshop* that suggests **constructing logic** to customize simulation of transport phenomena, nor is there any suggestion in *Real-Time Workshop* of **converting constructed logic** into object-oriented code.

The following schematic generally shows the relationship between Simulink, MATLAB, and *Real-Time Workshop* (which represents Applicants' understanding of *Real-Time Workshop* - see for example, page xvi of *Real-Time Workshop*).



As depicted above, *Real-Time Workshop* takes procedural logic embodied in Simulink and/or MATLAB models, and optionally constrained by scenarios prescribed in the Stateflow state machine, generates procedural C code, which can then be compiled and executed. The steps disclosed in *Real-Time Workshop* (1) do not take place without intervention of the user, much less does it occur in a single run as in Applicants' computer system and (2) the resulting code is procedural (not object-oriented), and without any teaching or suggestion of integration between data models.

Stroustrup discusses the capabilities and benefits of C++ programming language and promotes use of C++. *Stroustrup* states that C++ can support object-oriented programming, pointing out the advantages of C++ over other programming languages that support object-oriented programming such as Smalltalk and CLOS.

Some of the limitations of Applicants' claim 1 and the Examiner's assertions in the final Office Action are set forth below in a side-by-side presentation for ease of reference.

Applicants' Claimed Subject Matter	**Examiner's Assertions in final Office Action**
"convert the constructed logic into corresponding object-oriented code during a simulation without intervention of the simulator user"	"*automatically builds programs . . .*" (page 1-2, first paragraph of *Real-Time Workshop*; see page 5 of final Office Action)

Real-Time Workshop's statement that its simulation "automatically builds programs" can not refer to building object-oriented code since *Real-Time Workshop* does not generate object-oriented code. *Real-Time Workshop* can generate ANSI/ISO C code (among other languages), which can be compiled by C or C++ compilers. C is suited for procedural code, and does not naturally support object-orientation. C++ code by it nature supports object-oriented principles, however, code written in C++ could be plain procedural code, and C++ code is not necessarily object-oriented. Whether code is object-oriented is determined not by the compiler that processes the code, but whether object-oriented principles are employed. *Real-Time Workshop* does not generate code that is based upon object-oriented principles, and the integration between the components of its model is therefore fundamentally different from Applicants' claimed invention.

Applicants' Claimed Subject Matter	**Examiner's Assertions in final Office Action**
"integrate, without intervention of the simulator user, the object-oriented code with the main simulation system which comprises a simulation data model and simulation algorithms, resulting in an integrated simulation system . . ."	"*Seamless integration with MATLAB and Simulink*" (page 1-3, bulleted list), which the Examiner asserts comprises a simulation data model and simulation algorithms. "*how the generated model code is executed. The Real-Time Workshop generates algorithmic code as defined by your model. You may include your own code into your model via S-functions.*" (page 6-4, first paragraph). "*solver and data logging routines*" (page 6-4, figure 6-1, i.e. solver (algorithms) and data logs (data model) are present in the main simulation system, accessed via the Run-Time Interface. (Quoted passages are from *Real-Time Workshop*; see page 5 of final Office Action)
"wherein the object-oriented code extends the simulation data model by creating new classes that inherit from the simulation data model, and the object-oriented code is configured to call functions of the integrated simulation system and use member data of the integrated simulation system"	"*An open and extensible architecture*" (page 1-3, bulleted list); "*Because Simulink is customizable, you can further simplify modeling by creating custom blocks and block libraries from continuous and discrete-time components.*" (page 1-10, fourth paragraph); "*All Simulink blocks are automatically converted to code, with the exception of MATLAB function blocks and S-function blocks that invoke M-files.*" (page 1-4, second paragraph); and "solver and data logging routines" are accessed via the Run-Time Interface (see above). (Quoted passages are from *Real-Time Workshop*; see pages 5-6 of final Office Action)

Applicants' claimed subject matter integrates a model of a physical system (e.g., a facilities system) with the main simulation system, which is an existing system. One critical point is that the integration goes well beyond invoking code and passing data. The claimed subject matter integrates in a more profound way by <u>extending</u> the existing main simulation system <u>during the simulation and without intervention of the simulator user</u>. The simulation data model is extended by creating

new classes that inherit from the main simulation system, and the object-oriented code is configured to call functions of the integrated simulation system and use member data of the integrated simulation system. Amazingly, this level of integration is achieved at runtime, without intervention of the simulator user.

There is no teaching or suggestion in *Real-Time Workshop* of extending the simulation data model by creating new classes, which can only be done based upon object-oriented principles. Although *Real-Time Workshop* refers to the simulation as having "extensible architecture" and the *Real-Time Workshop* simulation is stated as being "customizable", these phrases in no way suggest object-oriented functionality. This flexibility available to *Real-Time Workshop* users refers to *Real-Time Workshop's* ability to invoke additional components, and does not refer to the type of flexibility in Applicant's claimed subject matter, because *Real-Time Workshop* can not teach characteristics of an object-oriented system since it does not generate object-oriented code.

The Examiner's reference to MATLAB is misplaced because MATLAB is a programming language for technical computing; MATLAB is not a simulator. The Examiner's assertion that the phrase "*seamless integration with MATLAB and Real-Time Workshop*" comprises a simulation data model and simulation algorithms is also misplaced because the term "data model" is defined in the present application to mean "a collection of C++ classes" (page 26, line 13), which are based upon object-oriented principles, which as discussed above, *Real-Time Workshop* does not support.

There is no teaching or suggestion in *Real-Time Workshop* of using object-oriented code. The Examiner's comments in the final Office Action suggest that he believes that *Real-Time Workshop* has the functionality of the claimed subject matter, and that one skilled in the art could read *Stroustrup* and rewrite *Real-Time Workshop* to be object-oriented. Since *Real-Time Workshop* does not disclose the fundamental capabilities of Applicants' claimed subject matter, persons skilled in the art would not take the *Real-Time Workshop* disclosure and combine it with *Stroustrup* to arrive at Applicants' claimed subject matter. Applicants' claimed invention automatically, and without simulator user intervention, takes advantage of sophisticated object-oriented techniques.

Simply put, neither *Real-Time Workshop* nor *Stroustrup* teach or suggest how to integrate, without intervention of the simulator user, the object-oriented code with the main simulation system and doing so during a simulation without intervention of the simulator user. At best, *Stroustrup* states that traditional design methods (non-object-oriented) are less resilient to change, less amendable to tools, less suited for parallel development, and less suited for concurrent execution. This, however, is not teaching or suggesting the elements of Applicants' claimed subject matter, which are not disclosed in *Stroustrup* or *Real-Time Workshop*, either alone or in combination.

For at least the above reasons, Applicants respectfully assert that the 35 U.S.C. §103 rejection of claim 1 is improper and should be overruled. For at least analogous reasons, Applicants also submit that the Examiner has not established a prima facie case of obviousness with respect to independent claim 20 which embodies similar limitations to claim 1 (with respect to the outstanding 35 U.S.C. §103 rejections). Further, Applicants submit that independent claim 1 and 20 and dependent claims 2-13, 15-19, 21-28, 30-31, and 43-46 are allowable.

FEES

Please charge ExxonMobil Deposit Account No. 05-1328 in the amount of $510.00 for submission of a Brief in Support of an Appeal and the amount $460 for a two-month extension fee. No additional fees or expenses are believed to be required; however, if any additional fees are required, please charge ExxonMobil Deposit Account No. 05-1328.

CONCLUSION

Based on the foregoing discussion, Applicants respectfully request that the Examiner's final rejections of claims 1-13, 15-28, 30-31 and 43-46 be overruled and withdrawn by the Board, and that the application be allowed to issue as a patent with all pending claims.

Respectfully submitted,

Date: 30 October 2007

Gary D. Lawson

Gary D. Lawson
Attorney for Appellants
Reg. No. 27,696

ExxonMobil Upstream Research Company
P.O. Box 2189
Houston, Texas 77252-2189
Tel. No. (713) 431-4846
Fax No. (713) 431-4664

CLAIMS APPENDIX

1. (*Previously presented*) A computer system for simulating a physical system comprising:

a processor;

memory coupled to the processor; and

object-oriented software in a main simulation system stored in the memory, the object-oriented software configured to:

a) provide a logic interface to dynamically construct logic to customize simulation of transport phenomena through a model of the physical system;

b) convert the constructed logic into corresponding object-oriented code during a simulation without intervention of the simulator user;

c) integrate, without intervention of the simulator user, the object-oriented code with the main simulation system which comprises a simulation data model and simulation algorithms, resulting in an integrated simulation system, wherein the object-oriented code extends the simulation data model by creating new classes that inherit from the simulation data model, and the object-oriented code is configured to call functions of the integrated simulation system and use member data of the integrated simulation system; and

d) execute the integrated simulation system.

2. (*Previously presented*) The computer system of claim 1 wherein the constructed logic comprises facility management logic which is representative of steps used to simulate the monitoring and controlling of mechanical facilities associated with the physical system.

3. (*Original*) The computer system of claim 1 wherein the logic interface comprises a logic flow chart interface.

4. *(Previously presented)* The computer system of claim 3 wherein the logic flow chart interface comprises one or more of icons, arrows, menus, dialogs, toolbar buttons, and text to enable the simulator user of the computer system to build-up, edit and visualize facility management logic in the form of a flow chart.

5. *(Original)* The computer system of claim 3 wherein the logic flow chart interface comprises icons representing basic logic control constructs for looping, decision making, statement execution, and logic entry and exit.

6. *(Previously presented)* The computer system of claim 5 wherein icons that represent logic control mechanisms enable the simulator user of the computer system to construct customized logic flow charts.

7. *(Original)* The computer system of claim 1 wherein the logic interface comprises a text-based logic code interface.

8. *(Original)* The computer system of claim 7 wherein the text-based logic code interface comprises a graphical text editor for performing one or more of entering, modifying and deleting lines of alpha-numeric text.

9. *(Previously presented)* The computer system of claim 7 wherein the text-based logic code is a facility management control language automatically created from a logic flow chart.

10. *(Previously presented)* The computer system of claim 9 wherein the facility management control language is automatically converted into object-oriented-facility management code in the form of C++.

11. *(Previously presented)* The computer system of claim 9 wherein the facility management control language is an object-oriented language that is parsed prior to conversion into the object-oriented-facility management code to verify syntax.

12. *(Previously presented)* The computer system of claim 1 wherein the object-oriented code is facility management object-oriented code in the form of C++.

13. *(Previously presented)* The computer system of claim 1 wherein the logic interface is configured to develop logic using either a logic flow chart interface or a text-based logic code interface.

14. (*Cancelled*)

15. (*Previously presented*) The computer system of claim 1 wherein the object-oriented software is further configured to compile the object-oriented code into object-oriented facility management object code, to link the object-oriented facility management object code to produce shared libraries, and to link the shared libraries into the main simulation system.

16. (*Previously presented*) The computer system of claim 1 wherein the object-oriented software is further configured to compile the object-oriented code into object-oriented facility management object code, to link the object-oriented facility management object code to produce dynamic linked libraries, and to combine the dynamic linked libraries with the main simulation system.

17. (*Previously presented*) The computer system of claim 1 wherein the object-oriented software is configured to execute the integrated simulation system by invoking the object-oriented facility management code at a plurality of timesteps during the simulation.

18. (*Previously presented*) The computer system of claim 17 wherein the object-oriented facility management code returns control back to the main simulation system after the facility management code has finished executing for a current timestep.

19. (*Previously presented*) The computer system of claim 1 wherein the processor comprises a plurality of connected processors to perform the simulation.

20. (*Previously presented*) A computer-implemented method of simulating a physical system comprising the steps of:

a) dynamically constructing logic to customize simulation of transport phenomena through a model of the physical system by a reservoir simulator user;

b) initiating simulation of transport phenomena through the model of the physical system by the reservoir simulator user causing initiation of the following steps without intervention of the reservoir simulator user:

i) automatically converting the logic into corresponding object-oriented code;

ii) integrating the object-oriented code with a main simulation system which comprises a simulation data model and simulation algorithms, resulting in an integrated simulation system for simulating the physical system, wherein the converted object-oriented code extends the simulation data model by creating new classes that inherit from the simulation data model, and the object-oriented code is configured to call functions of the integrated simulation system and use member data of the integrated simulation system; and

iii) executing the integrated simulation system to simulate the physical system.

21. (*Original*) The method of claim 20 wherein the physical system comprises a hydrocarbon-bearing subterranean formation.

22. (*Original*) The method of claim 21 wherein the physical system comprises fluid-containing facilities associated with production of hydrocarbons from the hydrocarbon-bearing subterranean formation.

23. (*Original*) The method of claim 20 wherein construction of the logic comprises using a graphical user interface to perform at least one step chosen from:

a) selecting and using an existing logic;

b) selecting and modifying existing logic; and

c) developing new logic.

24. (*Original*) The method of claim 23 wherein construction of the logic produces a logic flow chart.

25. (*Original*) The method of claim 23 wherein construction of the logic produces a text-based logic code.

26. (*Previously presented*) The method of claim 24 wherein construction of the logic flow chart comprises using one or more of icons, arrows, menus, dialogs, toolbar buttons, and text to enable the reservoir simulator user of the computer system to build-up, edit and visualize facility management logic in the form of a flow chart.

27. (*Original*) The method of claim 25 wherein the construction of the logic flow chart comprises using text-based logic code interface comprising a graphical text editor useful for entering, modifying and deleting lines of alpha-numeric text.

28. (*Original*) The method of claim 20 wherein the conversion of the logic is to C++ code.

29. (*Canceled*)

30. (*Original*) The method of claim 20 wherein execution of the initiated simulation system generates results for predicting the overall behavior of the physical system.

31. (*Original*) The method of claim 20 wherein execution of the initiated simulation system is carried out using a plurality of connected processors.

32-42. (*Cancelled*)

43. (*Previously presented*) The computer system of claim 1 wherein the object-oriented software is configured to construct the model representing facilities associated with the physical system and the constructed logic controls the operation of modeled facilities in the model at a plurality of timesteps during the simulation.

44. (*Previously presented*) The computer system of claim 43 wherein the mechanical facilities comprises fluid-containing facilities associated with production of hydrocarbons from a hydrocarbon-bearing subterranean formation.

45. (*Previously presented*) The computer system of claim 1 wherein the physical system comprises a hydrocarbon-bearing subterranean formation.

46. (*Previously presented*) The method of claim 20 comprising the steps of constructing the model that represents a reservoir and facilities associated with the physical system, wherein the constructed logic controls the operation of modeled facilities in the model at a plurality of timesteps during the simulation.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None